

polarzero

software engineer

Paris, France

[GitHub \(0xpolarzero\)](#) / [Twitter \(0xpolarzero\)](#) / contact@polarzero.xyz

Most of my experience as a SE is in AI and blockchain, and nearly all of my work has been open source, whether for a company, on contract, or independently. My current focus is on working sensibly with AI: refining my workflows as the core harness for doing good work, without getting one-shot by obsessive AI optimization or losing sight of architecture, design, product, and taste.

I'm interested in emergent capabilities, both in how frontier models behave and in how people evolve as they start using AI seriously. I try to keep as much skin in the game as possible by experimenting myself, discussing with peers I respect, and paying attention to early shifts in DevX, UX, tooling, interfaces, and product habits. Adjacent interests include local-first apps and capabilities, fully onchain games, and developer tooling.

The more I learn, the more the engineering side becomes intuitive, and the more I can think about product and intent. I'm trying to become more product-minded while keeping the discipline and curiosity to tackle unfamiliar problems with a good methodology.

WORK

2026-04
now

svvy

A strategic coding workbench for directing bounded, workflow-backed agent work

- **svvy**: organizes coding work around orchestrator sessions that hold product intent, route implementation into bounded threads, and reconcile durable results from structured, inspectable workflows those threads supervise without bloating orchestrator context, while letting you steer at any layer.
- **electrobun-browser-tools**: shipped an inspection and driving bridge for Electrobun apps, exposing windows, views, layout trees, DOM state, logs, events, screenshots, and Playwright-style locators to agents.
- **electrobun-e2e**: shipped shared end-to-end infrastructure for running Electrobun desktop apps headlessly in OrbStack Linux environments.
- Built with Electrobun, Svelte, Pi, and Smithers.

[github](#)

2025-04
now

Tevm

Contributing to a multi-language library for running an EVM in every environment

- **@tevm/compiler**: shipped a Solidity & Vyper compiler around Foundry compilers for Typescript.
- **tevm-monorepo**: contributed to call/debug and tracing methods, a MUD plugin for optimistic updates, storage layout and pre/post-state tooling, and various other runtime, API, build, and documentation contributions.
- **guillotine**: contributed to a Zig EVM, including a Zig devtool and BubbleTea CLI for call disassembly and step-by-step tracing, Go/C/WASM/TypeScript SDK bindings, EVM semantics fixes, and research-heavy work on hardfork support, gas accounting, and execution spec fixtures.
- **guillotine-mini**: contributed WASI/WASM build and bindings, and research-heavy EVM tracing/debugging work around WASM constraints, threading, debugger architecture, and dispatch-level execution hooks.
- **@tevm/test-matchers**: shipped a Javascript library that extends Vitest with EVM-related test matchers.
- **@tevm/test-node**: shipped a Javascript library to snapshot EVM JSON-RPC calls in Vitest/Bun.

[website](#) [github](#) [twitter](#)

2024-04
2025-03

Primodium

Worked at a startup backed by Alliance, Paradigm, and A16Z Games, exploring onchain games and crypto user-facing products

- **DEX Indexer**: shipped a Yellowstone gRPC Typescript indexer for Solana DEX trades.
- **DEX GraphQL**: shipped a Hasura + Timescale GraphQL client for querying DEX activity & analytics on Solana.
- **DEX Server**: contributed server/package work around buy/sell flows, SOL/USD price caching, transaction analytics, Hasura/cache integration, Docker/package workflows, and docs.
- **Tub**: contributed across the Solana indexer, GraphQL/Hasura/Timescale layer, dashboard/explorer, server analytics, and iOS query/chart/transaction integration surfaces.
- **Gasless server**: shipped a MUD-compatible gasless server library for EVM chains.
- **Primodium Empires**: contributed client UI/game tooling, cheatcodes, transaction feedback, keeper/deployment infrastructure, and contract test/audit-prep work, while owning the artist handoff loop for integrating art and animations into the game.
- **Reactive Tables**: shipped a state management library for onchain games built on MUD for Typescript & React.
- **Primodium v0.11**: took ownership of the sync/indexer and database stack, and shipped client/core rendering work, game-object interaction fixes, reactive-table integration, package/build fixes, and browser profiling-driven performance optimizations.
- **Open-source release**: owned the public release pass for the work above, including documentation for each package and shipping the open-sourced libraries and containers.

[website](#) [github](#) [twitter](#)

RESEARCH

2024-02
2024-03

Research: EVM gas benchmarks

Various research projects on EVM gas usage and tooling

- **airdrop gas benchmarks**: a series of tests to benchmark gas usage across ERC20/721/1155 patterns with batched, merkle, and claim style drops, picked from popular airdrop contracts — comes with an interactive dashboard to analyze costs based on airdrop parameters.
- **gas metering comparison**: cross-validated gas reports from popular tooling against live executions with Foundry, Hardhat, and Tevm on identical calldata sets, and documented discrepancies.

2023-11
2023-12

Research: EVM security

Various research projects on EVM security and tooling

- **Glider**: joined Secureum workshop sessions to battle-test Glider on live exploit scenarios, and submitted documentation fixes and clarified flows for security researchers.
- **storage collision**: a reference research for verifying smart contract assumptions using fuzzing & formal verification tools (here exhibiting storage collision) with Foundry, Halmos, and Certora.
- **ERC1155A**: a reference fuzzing test suite on a token extension to verify assumptions and surface edge cases.

More about my projects, experiments, writing and education at polarzero.xyz